CLAIMS

What is claimed is:

1. A classfile modification method, comprising:

converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;

adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,

converting the resulting collection of objects into a modified version of said classfile.

2. The classfile modification method of claim 1 wherein said collection of objects are partially defined by a first class that used as a template to generate said unique objects.

3. The classfile modification method of claim 2 wherein said modifying further comprises creating an object from said template that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a method that registers the identity of the class, and the methods of said class, that said classfile describes with a feature of another class.

4. The classfile modification method of claim 2 wherein said collection of objects are partially defined by a second class that is used as a template to generate a unique object for each field information structure found in said classfile.

5. The classfile modification method of claim 4 wherein said modifying further comprises creating a new object that represents a new field information structure for said classfile.

6. The classfile modification method of claim 5 wherein said field information structure is to store a numeric identification assigned to said class.

7. The classfile modification method of claim 1 wherein said unique object corresponds to a constructor method.

8. The classfile modification method of claim 1 wherein said adding at least one additional other object further comprises adding an additional other object at a

position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached.

9. The classfile modification method of claim 8 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

10. The classfile modification method of claim 8 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

11. The classfile modification method of claim 1 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached.

12. The classfile modification method of claim 11 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

13. The classfile modification method of claim 11 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

14. The classfile modification method of claim 1 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

15. The classfile modification method of claim 14 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

16. The classfile modification method of claim 14 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

17. The classfile modification method of claim 1 wherein said adding at least one additional other object further comprises:

adding a first additional other object at a position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached;

adding a second additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached; and, adding a third additional object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

18. The classfile modification method of claim 17 wherein said classfile is a Java compatible classfile and said first, second and third additional other objects correspond to the addition of invokestatic instructions.

19. The classfile modification method of claim 17 wherein said classfile is a Java compatible classfile and said first, second and third additional objects correspond to the addition of invokevirtual instructions.

20. A machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;

adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,

converting the resulting collection of objects into a modified version of said classfile.

21. The machine readable medium of claim 20 wherein said collection of objects are partially defined by a first class that used as a template to generate said unique objects.

22. The machine readable medium of claim 21 wherein said modifying further comprises creating an object from said template that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a method that registers the identity of the class, and the methods of said class, that said classfile describes with a feature of another class.

23. The machine readable medium of claim 21 wherein said collection of objects are partially defined by a second class that is used as a template to generate a unique object for each field information structure found in said classfile.

24. The machine readable medium of claim 23 wherein said modifying further comprises creating a new object that represents a new field information structure for said classfile.

25. The machine readable medium of claim 24 wherein said field information structure is to store a numeric identification assigned to said class.

26. The machine readable medium of claim 20 wherein said unique object corresponds to a constructor method.

27. The machine readable medium of claim 20 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached.

28. The machine readable medium of claim 27 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

29. The machine readable medium of claim 27 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

30. The machine readable medium of claim 20 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached.

31. The machine readable medium of claim 30 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

32. The machine readable medium of claim 30 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

33. The machine readable medium of claim 20 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

34. The machine readable medium of claim 33 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

35. The machine readable medium of claim 33 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

36. The machine readable medium of claim 20 wherein said adding at least one additional other object further comprises:

adding a first additional other object at a position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached;

adding a second additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached; and,

adding a third additional object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

37. The machine readable medium of claim 36 wherein said classfile is a Java compatible classfile and said first, second and third additional other objects correspond to the addition of invokestatic instructions.

38. The machine readable medium of claim 36 wherein said classfile is a Java compatible classfile and said first, second and third additional objects correspond to the addition of invokevirtual instructions.

39. A computing system implemented with a machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

converting said classfile into a collection of objects whose organization is derived from said classfile's organization, said collection of objects comprising unique objects for each method information structure found in said classfile, each of said unique objects referenced to other objects in said organization, said other objects representing their corresponding unique object's method information structure's byte code instructions;

adding at least one additional other object to an arrangement of other objects that are referenced to a unique object, said at least one additional other object corresponding to at least one byte code instruction that causes a plug-in module's handler method to provide output function treatment for said unique object's method; and,

converting the resulting collection of objects into a modified version of said classfile.

40. The machine readable medium of claim 39 wherein said collection of objects are partially defined by a first class that used as a template to generate said unique objects.

41. The machine readable medium of claim 40 wherein said modifying further comprises creating an object from said template that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a method that registers the identity of the class, and the methods of said class, that said classfile describes with a feature of another class.

42. The machine readable medium of claim 40 wherein said collection of objects are partially defined by a second class that is used as a template to generate a unique object for each field information structure found in said classfile.

43. The machine readable medium of claim 42 wherein said modifying further comprises creating a new object that represents a new field information structure for said classfile.

44. The machine readable medium of claim 43 wherein said field information structure is to store a numeric identification assigned to said class.

45. The machine readable medium of claim 39 wherein said unique object corresponds to a constructor method.

46. The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises adding an additional other object at a

position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached.

47. The machine readable medium of claim 46 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

48. The machine readable medium of claim 46 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

49. The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached.

50. The machine readable medium of claim 49 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

51. The machine readable medium of claim 49 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

52. The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

53. The machine readable medium of claim 52 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokestatic instruction.

54. The machine readable medium of claim 52 wherein said classfile is a Java compatible classfile and said additional other object corresponds to the addition of an invokevirtual instruction.

55. The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises:

adding a first additional other object at a position that corresponds to a region of said unique method's instructions that is executed just after said unique method's entry point is reached;

adding a second additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached; and, adding a third additional object at a position that corresponds to a region of said unique method's instructions that will be executed if an error arises during execution of said unique method.

56. The machine readable medium of claim 55 wherein said classfile is a Java compatible classfile and said first, second and third additional other objects correspond to the addition of invokestatic instructions.

57. The machine readable medium of claim 55 wherein said classfile is a Java compatible classfile and said first, second and third additional objects correspond to the addition of invokevirtual instructions.